

# Remote MIIC Administration

These instructions explain how to access the management & administration functions of a MIIC server.

MIIC uses the Java Management interface (JMX) to provide administration functionality. This requires launching some tool that connects to the MIIC Java VM via the JMX interface. Normally, this is done with the "jconsole" GUI.

Unfortunately, the jconsole GUI is not well suited to our needs:

- Although jconsole can attach to a remote Java VM this requires a special tomcat server configuration. We would prefer not to go down this route as it involves IT security, etc.
- Enabling a remote x-windows display of the jconsole GUI is both difficult to setup (i.e. the X-windows authentication tokens), and the resulting GUI is usually sluggish.

We will instead use a text-only JMX console called [TJConsole](#).

## How to use TJConsole

1. Download the [tjconsole jar file](#)
2. Find the java PID

```
ps -elf | grep java
```

3. Launch *tjconsole* and connect to the java PID. This command must run as user "tomcat"

```
sudo -u tomcat java -jar ./tjconsole-1.7-all.jar -xterm -connect [PID]
```

4. Connect to a management bean (MBean) and issue commands. For example, use the opendapLoader MBean to clear local OPeNDAP cache:

```
TJConsole>use bean:name=opendapLoader
Attached to bean bean:name=opendapLoader
TJConsole>invoke clearCache

Method result:(void)
```

## MIIC Management Interface

MBean	Function	Description
bean:name=opendapLoader	clearCache()	Destroys all local OPeNDAP cache.  Local cache includes locations of data collection files, and the history of scanned OPeNDAP folders. After calling this method, the first few times you locating data collection files may run more slowly than normal.
bean:name=opendapLoader	getStatusMessage()	Retrieves the status of OPeNDAP servers.  The empty string is returned for normal status.

bean:name=opendapLoader	rescanServers()	Removes history of scanned OPeNDAP folders.  Normally MIIC will not scan OPeNDAP folders it has already scanned in order to save time. Call this function if the contents of the server have changed and you wish to scan again.
bean:name=opendapLoader	getMaxOPeNDAPFetch()	Returns the number of concurrent OPeNDAP requests.
bean:name=opendapLoader	setMaxOPeNDAPFetch(max:int)	Sets the number of concurrent OPeNDAP requests.  Do not set this to be larger than the max number of OPeNDAP beslistener processes.
bean:name=opendapLoader	verifyServers()	Verify that OPeNDAP servers are available and have the MIIC plugin installed